PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : H04L 29/08, G06F 15/02, H04L 29/06 — A1

(11) International Publication Number: WO 99/39488

(43) International Publication Date: 5 August 1999 (05.08.99)

(21) International Application Number: PCT/GB99/00311

(22) International Filing Date: 29 January 1999 (29.01.99)

(30) Priority Data:

98300658.6	29 January 1998 (29.01.98)	EP
9801928.4	29 January 1998 (29.01.98)	GB

(71) Applicant *(for all designated States except US)*: BRITISH TELECOMMUNICATIONS PUBLIC LIMITED COMPANY [GB/GB]; 81 Newgate Street, London EC1A 7AJ (GB).

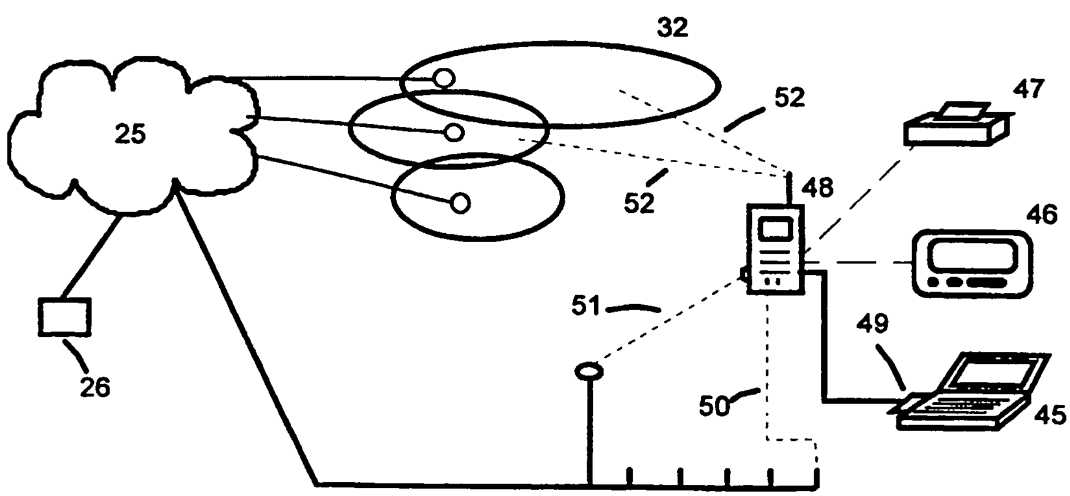
(72) Inventor; and
(75) Inventor/Applicant *(for US only)*: FLYNN, Jason, Stuart [GB/GB]; 32 Church Road, Felixstowe, Suffolk IP11 9NF (GB).

(74) Agents: READ, Matthew, Charles et al.; Venner, Shipley & Co., 20 Little Britain, London EC1A 7DH (GB).

(81) Designated States: CN, IN, JP, SG, US, European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

Published
With international search report.
Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(54) Title: COMMUNICATIONS SYSTEM FOR MOBILE DATA TRANSFER



(57) Abstract

A communications system for mobile data transfer, comprising a communications network (25), an application module (45, 46, 47) and a separate terminal module (48), where data is transferred between the network (25) and the application module (45, 46, 47) via the terminal module (48), which hides network dependent operations such as error control from the application module. By implementing the communications stack (12, 13) in the terminal module rather than in the application module, the invention is capable of providing a complete communications package, in a portable terminal module (48) or on a single plug-in card, capable of interfacing with a wide variety of application modules (45, 46, 47, 92).

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania
AM	Armenia
AT	Austria
AU	Australia
AZ	Azerbaijan
BA	Bosnia and Herzegovina
BB	Barbados
BE	Belgium
BF	Burkina Faso
BG	Bulgaria
BJ	Benin
BR	Brazil
BY	Belarus
CA	Canada
CF	Central African Republic
CG	Congo
CH	Switzerland
CI	Côte d'Ivoire
CM	Cameroon
CN	China
CU	Cuba
CZ	Czech Republic
DE	Germany
DK	Denmark
EE	Estonia
ES	Spain
FI	Finland
FR	France
GA	Gabon
GB	United Kingdom
GE	Georgia
GH	Ghana
GN	Guinea
GR	Greece
HU	Hungary
IE	Ireland
IL	Israel
IS	Iceland
IT	Italy
JP	Japan
KE	Kenya
KG	Kyrgyzstan
KP	Democratic People's Republic of Korea
KR	Republic of Korea
KZ	Kazakstan
LC	Saint Lucia
LI	Liechtenstein
LK	Sri Lanka
LR	Liberia
LS	Lesotho
LT	Lithuania
LU	Luxembourg
LV	Latvia
MC	Monaco
MD	Republic of Moldova
MG	Madagascar
MK	The former Yugoslav Republic of Macedonia
ML	Mali
MN	Mongolia
MR	Mauritania
MW	Malawi
MX	Mexico
NE	Niger
NL	Netherlands
NO	Norway
NZ	New Zealand
PL	Poland
PT	Portugal
RO	Romania
RU	Russian Federation
SD	Sudan
SE	Sweden
SG	Singapore
SI	Slovenia
SK	Slovakia
SN	Senegal
SZ	Swaziland
TD	Chad
TG	Togo
TJ	Tajikistan
TM	Turkmenistan
TR	Turkey
TT	Trinidad and Tobago
UA	Ukraine
UG	Uganda
US	United States of America
UZ	Uzbekistan
VN	Viet Nam
YU	Yugoslavia
ZW	Zimbabwe

COMMUNICATIONS SYSTEM FOR MOBILE DATA TRANSFER

Field of the Invention

This invention relates to a communications system, with particular but not exclusive application in mobile data transfer over networks based on the Internet Protocol.

Background

Mobile data communications systems are well known. For example, to connect to the mesh of networks, or internetwork, known as the Internet, most mobile data users currently use a combination of a GSM mobile telephone together with a data card and suitable host platform, typically a notebook computer running one or more application programs such as e-mail or remote file transfer across the network. Generally, mobile systems require that the mobile host is reachable at the same address as it moves across different networks, with continuous connectivity and seamless roaming even while a network application is running.

The architecture of a generalised communications system may be conveniently described by reference to the well-known 7-layer International Standards Organisation (ISO) reference model, shown in Figure 1.

The model portrays a communications subsystem comprising two host computers A and B, each represented by seven protocol layers 1 to 7, connected through a data network 8 via a gateway or router 9. The arrows between the layers indicate the logical connections between them. Each layer in the model performs a well defined function and operates according to a defined protocol, which allows it to exchange messages with the corresponding layer in a remote system. In practice, this exchange is possible through data encapsulation: at the transmitting host, data generated by the user application process (AP) 10 is passed down the layer stack as a series of

data units, with each layer adding control information to the units until they are in a form capable of being sent across the network. At the receiving host, each received data unit passes up the layer stack towards the user AP 10, each receiver layer stripping out the control information produced by the corresponding transmitter layer.

Layers 1 to 3 are said to implement network dependent functions, the detailed operation of which varies between different network types. Layer 1, the physical layer, provides the physical, for example, electrical means for transmitting a serial bit stream across the network 8. Layers 5 to 7 are said to implement application oriented or network independent functions, with layer 7, the application layer, providing the user interface. Layer 4, the transport layer, acts as an interface between the network dependent and the application oriented layers, providing the session layer 5 above it with a message transport facility which is independent of the underlying network. The transport layer is responsible for end-to-end message transfer between Host A and Host B, including functions such as connection management and error control. Reference is directed to "Data Communications, Computer Networks and Open Systems", Fred Halsall, Addison-Wesley, 4th Edition, Chapter 1.3, for a detailed description of the ISO reference model.

Within the Internet, data transfer is based on a set of protocols generically referred to as TCP/IP, which provides a layered protocol architecture commonly referred to as a protocol or communications stack. As well as basic network protocols, the TCP/IP protocol suite includes a number of well-known application protocols, including the File Transfer Protocol (FTP) and the Remote Terminal Protocol (TELNET). A simplified diagram of the logical structure of these layered protocols is shown in Figure 2.

The TCP/IP suite is generally described on the basis of a 4-layer model, rather than the 7-layer ISO model of Figure 1, although there is general correspondence between the two models. The application protocol layer 11

encompasses layers 5 to 7 in the 7-layer model and provide functions such as remote file transfer (FTP), remote terminal login (TELNET) and e-mail (SMTP). The TCP/UDP (Transmission Control Protocol/User Datagram Protocol) layer 12 corresponds to layer 4 in the 7-layer model. TCP and UDP are transport protocols concerned with the management of end-to-end message transfer between two hosts. The IP (Internet Protocol) layer 13 corresponds generally to layer 3 and the Network Interface layer 14 corresponds to layers 1 and 2 in the 7-layer model, and provides the means for converting between data units at the IP layer and the data format required for transmission over the physical network 8, for example, Ethernet frames for an Ethernet based network.

The Internet Protocol (IP) provides a number of core functions and associated procedures necessary when working across dissimilar networks, including fragmentation and re-assembly of user messages, network routing and addressing. Data is transferred in the form of data units known as IP datagrams between points in the Internet specified by IP addresses. The detailed specification of IP is available in a "Request for Comments" document, RFC 791, maintained by the Internet Engineering Task Force (IETF), which is widely available on the Internet at, for example, "ftp://ds.internic.net/rfc/rfc791.txt". As well as being central to Internet technology, IP is widely used as the basic protocol for data transfer in internetworks outside of the Internet.

Data transfer within the Internet may take place on a connectionless or connection-oriented basis, depending on the protocol used. User Datagram Protocol (UDP) provides a connectionless IP datagram delivery service which does not maintain an end-to-end connection between transmitting and receiving hosts and therefore does not guarantee delivery. It merely treats each datagram as a self-contained entity to be transferred on a best-try approach. It is up to the application using the UDP service to perform error checking: if it does not do so, it has no way of knowing if a datagram has

arrived at the receiver or if it has been lost in transit. This form of transfer is suitable for some types of data, for example image or voice data, where speed may be more important than occasional errors, which are unlikely to substantially affect the received image or speech quality.

However, for most applications, a reliable connection-oriented service is required, which guarantees IP datagram delivery. The Transmission Control Protocol (TCP) is a connection-oriented protocol which maintains an end-to-end connection between pairs of communicating processes running on host computers, and provides a reliable and secure logical connection for data transfer between the processes.

TCP assumes that it can obtain a simple but potentially unreliable data service from the IP layer below it. To turn this into a reliable service, it therefore has to provide a range of functions including (a) basic data transfer, (b) reliability and error correction, allowing recovery from damaged or missing data or data delivered out of sequence, (c) flow control, which provides the receiving host with the means to govern the amount of data sent by the transmitting host, (d) multiplexing, which allows many processes within a single host to use its TCP facilities simultaneously, (e) maintenance of connection information for each connection between two processes, and (f) precedence and security. The TCP specification setting out the detailed requirements for implementation in each of these areas is available as RFC 793.

In known systems, the TCP/IP protocol stack is implemented in software, and is normally resident in, and an integral part of, the operating system of an application module, for example, a notebook computer. User access to the TCP through the operating system is comparable to similar processes such as user access to the computer's file system. A variety of TCP/IP implementations are commercially available for different platforms such as DOS and UNIX, for example, Microsoft TCP/IP software is provided as an

integral part of the Microsoft Windows 95 and Windows NT operating systems.

Referring to Figure 3, the way in which TCP operates to transfer data between two host computers may be illustrated by reference to the well-known client-server model. For example, one of the fundamental requirements in distributed systems is for access to information stored on remote file servers. FTP provides such access. Client A may be an application module such as a notebook computer and the remote computer B from which it requests information is known as the server. Both client and server operating systems contain TCP/IP modules.

The user accesses the client FTP process 16 by issuing commands through the client terminal 17. Alternatively, a user program/ application process (AP) 17 may access FTP. In either case, the user commands are passed to the client operating system 18 which calls the client FTP 16. The client FTP calls the TCP module 19 via the operating system 18. The client FTP 16 uses the TCP data transfer service between client and server TCP modules 19, 20 to send the commands to the server FTP 21. The server FTP then accesses its local file system 22 through its operating system 23, according to the commands specified by the user, and transmits the required data back to the client FTP 16 for presentation to the user. The way in which TCP establishes connections to the remote server is described below.

To identify the separate data streams that a TCP implementation within a host can handle, the TCP 19 provides a port identifier 24 for each separate process. Port identifiers are selected independently by each TCP and so may not be unique. To provide for unique addresses, the port identifier is combined with the IP address of the host to create a socket which will be unique throughout all networks connected together. A connection is fully specified by a pair of sockets at each end, one at the client TCP 19 and one at the server TCP 20. Each host socket may participate in many separate

connections.

The sockets model was originally developed by Berkeley Software Distribution (BSD) at the University of California at Berkeley and is the standard on which all TCP/IP software is based, with the aim of providing a common interface for network use on any particular operating system. In the case of Microsoft Windows based platforms, an applications programming interface (API) known as Microsoft Winsock (Windows sockets) implements a Windows based version of the BSD sockets model. Winsock software is installed alongside TCP/IP in Windows based systems to permit the interoperability of applications and TCP/IP implementations written to comply with the Winsock specification. The Microsoft Winsock specification is available on the Internet at a variety of sites, including for example: ftp://ftp.microsoft.com/bussys/Winsock/spec11.txt.

The port identifier forming part of the socket is pre-defined for a number of well-known processes such as FTP and TELNET: for example, any server implementing the FTP process is given the port number 21, so that the FTP process running on any server may be accessed simply by connecting to port number 21 of that server. Reference is directed to the TCP specification at RFC 793 cited above for a description of the way in which sockets are established, maintained and closed.

As a result of the extensive functionality provided by TCP, one problem associated with its use is the large overhead resulting from the implementation of the full TCP/IP protocol stack, both in terms of the memory required and because the additional processing performed by TCP, as required for most network applications, substantially increases the processing load on the client processor.

Summary of the Invention

To address the above problems, the present invention provides a

communications system for mobile data transfer, comprising a communications network, an application module capable of processing network data, and a separate terminal module configured to provide a network independent data transport service to the application module.

Network data may be data which originates at the network and is destined for the application module or vice versa.

The system according to the invention appreciates that the complexity of the protocol stack may be transferred from each of a plurality of application modules to a separate terminal module. The communications system may then comprise a single complex terminal module and one or more relatively simple application modules. This enables the construction of a variety of cheap application modules with limited functionality, and therefore a reduced level of complexity and cost compared with a conventional system, since there is no need to provide the memory space and processing power for sophisticated data transfer management, which is dealt with by the terminal module.

The removal of the protocol stack into a separate device also enables the use of a wider variety of application modules with the system, where application modules were previously incapable of implementing the complexity of the TCP/IP protocol stack.

In a system according to the invention, the application module may be configured so that data transfer between the network and the application module can only take place via the terminal module.

Data transfer between the network and the terminal module does not involve any processing within the application module, so that high overhead operations such as error checking do not place a load on the application module operating system or processor. These operations may be carried out

by a processing unit within the terminal module. When data transfer to the application module is required, that transfer may be carried out between the terminal module and the application module across hardware links, with very low bit error rates.

The hardware links may be in the form of physical wiring, for example, utilising the existing serial ports on a standard computer, or may be infra-red links. Various other forms of link including optical fibre connections are also envisaged.

As well as error control, the terminal module may be used to carry out all connection management, data fragmentation and flow control tasks, so hiding the detailed operation of the underlying network from the application modules.

The separation of the protocol stack from the application modules means that no protocol is defined for transfer of data between the terminal module and each application module. A protocol must therefore be specified to ensure that application processes running in all types of application modules should be able to access the stack in a uniform manner. The present invention accordingly further provides that the terminal module and the application module each include means configured to provide a logical connection between them.

The load on the application module processor can be lightened further by incorporating into the terminal module at least some of the functionality of the session and presentation layers in the general 7-layer model. For example, network data can be converted into standard ASCII format by the terminal module prior to being communicated to the application modules. The disadvantage of this approach is that greater functionality at the terminal module can lead to a reduction in flexibility at the application modules.

According to a second aspect of the invention, there is provided a terminal module for a mobile communications network, comprising a first connecting means for connecting the terminal module to the network and a second connecting means for connecting the terminal module to a separate application module, wherein the terminal module is configured to provide a network independent data transport service to each of a plurality of separate application modules.

Preferably, the terminal module is in a form and of a size which may be conveniently carried by the user at all times, for example mounted on the user's belt. The device may be incorporated into a PCMCIA-style card, which is easily carried and which may be conveniently connected to an application module by plugging the card into a corresponding slot in the module. As a result, this invention can provide a complete communications package on a single plug-in card.

The invention further provides a method of transferring data to an application module in a communications network, comprising receiving data destined for the application module at a terminal module separate from the application module and processing the received data so as to provide a network independent data transport service for the application module.

Brief Description of the Drawings

Embodiments of the invention will now be described by way of example, with reference to the accompanying drawings, in which:

Figure 1 shows the 7-layer ISO reference model for a general communications subsystem;

Figure 2 is a schematic diagram showing the protocol architecture of the protocols within the TCP/IP protocol suite;

Figure 3 is a schematic diagram showing the client/server model applied to an FTP process within the Internet;

Figure 4 is a schematic diagram showing a conventional data communications system;

Figure 5 is a flow diagram illustrating the operation of the system of Figure 4;

Figure 6 is a schematic diagram showing a system according to the present invention;

Figure 7 is a schematic representation of the terminal module and an application module;

Figure 8 is a flow diagram illustrating the operation of the system of Figure 6; and

Figure 9 illustrates a practical application of a system according to the present invention.

Detailed Description

Referring to Figure 4, a conventional data communications system comprises an IP based network 25 connected to the Internet (not shown) via a gateway 26. A plurality of application modules including notebook computers 27, 28 and an integrated palmtop computer/ mobile telephone 29, such as the Nokia 9000 Communicator, may be connected to the network 25. For example, a notebook computer 27 is connected to the network 25 by means of a data card 30 and GSM mobile telephone 31 through mobile IP based RF cells 32. The connection is typically established through an Internet Service Provider (ISP) which provides access to the Internet. Other application modules such as the notebook computer 28 may connect to the Internet through a hard-wired connection 33, for example through the computer's standard serial port, or by an infra-red link 34 using an infra-red port on the notebook to connect to an infra-red port 35 on the network 25. Each application module 27, 28, 29 has a TCP/IP communications stack installed as part of its local operating system and allows communication to be established with an Internet based host (not shown) through one or more reliable TCP connections.

The operation of the conventional system may be understood by reference to an application process, for example a user using a notebook computer with

FTP and TCP/IP installed, accessing a remote file system, as shown schematically in Figure 5.

For clarity and simplicity, it is assumed that the client FTP knows the IP address of the destination server, an operation which would normally be carried out separately by reference to a host computer known as a domain name server. Referring to Figure 5, the user requests access to a remote file on the file server 40 by keying in commands at the application module terminal 41, specifying the server location and the file name. The operating system (not shown) issues an "f_open" request to the client FTP 42 (step s1). In turn, the client FTP 42 issues an ACTIVE_OPEN request to the client TCP module 43 (s2) specifying the port identifier (21 is the well-known port for FTP) and the IP address of the server 40. The client TCP 43 responds with an OPEN_ID message (s3) informing the client FTP 42 of the local connection name which TCP has assigned to this connection. The client TCP 43 then seeks to establish the connection to the server 40 (s4 - s9). The sequence of protocol commands through which the TCP modules 43, 44 co-operate to establish the connection and the subsequent data transfer is well-known and is fully documented in the TCP specification (RFC 793). Reference is further directed to "Data Communications, Computer Networks and Open Systems", Fred Halsall, Addison-Wesley, 4th Edition, Chapters 11, 13 and 14 for an overview of the process.

Referring to Figure 6, a system according to the invention comprises a network 25 connected to the Internet (not shown) via a gateway 26, a number of application modules 45, 46, 47, and a separate terminal module 48. The application modules 45, 46, 47 are not directly connectable to the network 25 but are connectable to the terminal module 48. The terminal module 48 is in turn connectable to the network 25. In contrast to the application modules 27, 28, 29 in the conventional system, application module 45, 46, 47 in a system according to the invention do not contain a communications stack, which is implemented only in the separate terminal module 48.

Each application module 45, 46, 47 may be connected to the terminal module 48 in a number of different ways. For example, the terminal module 48 and the application module 46 each include an infra-red port (not shown), comprising an infra-red transmitter and receiver, and may therefore transmit and receive data conforming to the IrDA (Infra-red Data Association) standard. IrDA compliant devices within application modules are becoming increasingly common; for example, the Psion 3c palmtop computer includes an infra-red port, as do the Gateway Solo range of notebook computers, from Gateway 2000 Europe, Dublin, Ireland. The provision of an infra-red port allows a number of application modules to communicate with the terminal module 48 as long as each is within range.

Alternatively, or in addition, the terminal module 48 is provided with a PCMCIA-style card connector 49 which plugs into a corresponding slot in an application module 45, such as a notebook computer. Although it is possible to use a true PCMCIA card with existing application modules such as notebook computers, it is envisaged that a much simpler hardware based protocol be used over the terminal module to application module connection, as described in detail below. To ensure compatibility with existing notebook computers, the physical connection between terminal module 48 and application module 45 can be provided by using a card which meets the physical PCMCIA specification but not the electrical specification, referred to herein as a "PCMCIA-style" card. In an alternative embodiment, the terminal module 48 is entirely contained on a single PCMCIA-style plug-in card which may itself be plugged into the corresponding slot in an application module.

The terminal module 48 may be connected to the network 25 by a number of different means, including a direct wire link 50, an infra-red (I-R) link 51 and/or radio frequency (RF) links 52 via the mobile-IP based RF cells 32 of the network 25. Different options may be included in the device 48 according to the user's requirements. In the case of the infra-red link 51 and radio frequency link 52, the terminal module 48 and the network 25 each comprise

an appropriate transmitter and receiver (not shown).

Referring to Figure 7, the terminal module 48 comprises a network interface 60, a processing unit 61 and an operating system (OS) 62 which includes a number of modules. The modules are a TCP/IP module 63, a Sockets Interface module 64 and a Hardware Socket module 65. Finally, the device 48 includes an input/output interface 66.

The terminal module 48 is, for example, a conventional notebook computer, such as a Gateway Solo 9100. The network interface 60 is provided by a conventional GSM mobile telephone connected to the notebook 48 through a suitable data card, for example a Nokia PC card for Nokia 21 series GSM telephones. The notebook's integral infra-red port may also be used to access the network and a direct wire link is provided with a suitable network card, for example a 3Com Ethernet Combo PC card and Ethernet cable, assuming an Ethernet 10BASE-T network 25. The processing unit 61 is provided by the processor and associated circuitry of the notebook 48. The notebook 48 is pre-loaded with operating system 62 and suitable TCP/IP software to provide the TCP/IP module 63. For example, network access is provided on a notebook running Microsoft Windows 95 by configuring the pre-loaded Microsoft TCP/IP software and loading Microsoft Winsock software, which may be downloaded from the Internet at, for example, ftp://ftp.microsoft.com /bussys/WinSock/winsock2. The notebook 48 uses its infra-red port 66 for communications with the application modules 45, 46, 47. The Sockets Interface 64 and Hardware Socket 65 are software modules designed to reside in the operating system 62, the functionality of which is described in detail below.

Each application module 45, 46, 47 comprises a processor 67 capable of running an application process 68, an input/output port 69 for connection to the terminal module 48 and an operating system (OS) 70 including a Dummy Sockets Interface module 71 and a Hardware Socket module 72. The

input/output port 69 includes the facility for a direct wired connection, an infra-red transmitter/receiver and/or plug-in card connector. A generalised application module 45 is defined by reference to a Gateway Solo 9100 notebook computer, as for the terminal module 48. This is provided with a processing unit to run an application process, for example an FTP program and an infra-red transceiver, as described above. The Dummy Sockets Interface 71 and Hardware Socket 72 are software modules designed to reside in the operating system 70, as described in detail below.

The combination of the terminal module 48 and a single application module 45 may be considered to define a communications subsystem. Referring to Figure 2, the terminal module 48 implements layers 12, 13 and 14, corresponding to layers 1 to 4 in the 7-layer model. The application module implements layer 11, corresponding to layers 5 to 7 in the 7-layer model. The connection between the terminal module 48 and the application module 45 completes the communication subsystem.

To allow comparison with the conventional system, the operation of the system according to the invention will be described in terms of a request by a client notebook computer for transfer of a data file from a remote server.

Referring to Figure 8, a user at the notebook terminal 80 enters a request for a remote file from the file server 81, specifying the name of the server and the file name. As with the conventional system described at Figure 5, it is assumed that the client FTP 82 has already obtained the IP address of the server. The local operating system issues an "f_open" request to the client FTP 82 (s11), which issues an ACTIVE_OPEN command (s12) which is passed by the operating system (not shown) to the address at which TCP/IP would normally be resident. Since TCP/IP is not present, the Dummy Sockets Interface module 71 is located at the TCP address and is designed to present the same front-end to the operating system and therefore to accept the operating system call as if it were the TCP/IP module. This ensures that

existing applications such as FTP and Telnet which generate standard calls to TCP do not require modification to be used with the system according to the invention. The function of the Dummy Sockets Interface 71 is to convert the operating system call into a form which can be used by the Hardware Socket 72 to communicate with the Hardware Socket 65 in the terminal module 48. The Dummy Sockets Interface 71 therefore implements a look-up table which contains a unique token corresponding to each operating system call. The token for the ACTIVE_OPEN command may simply be a number, for example decimal "10".

The respective Hardware Sockets 72, 65 communicate by passing such tokens (s13), which may take a variety of physical forms depending on the nature of the input/output interface 66, 69. For example, the token may be represented as a binary sequence on bus lines or a voltage level on an analog line or a burst of light on a particular frequency over an optical link. In the case of an infra-red link, transmission of a token can be implemented by direct modulation of an infra-red emitter, using on-off keying (OOK), with binary 1 turning the emitter on and binary 0 turning it off. Reference is directed to "Data Communications, Computer Networks and Open Systems", Fred Halsall, Addison-Wesley, 4th Edition, Chapter 6, pages 332 - 334 for a detailed description of infra-red modulation techniques.

Assuming that the devices make use of the infra-red capability of the respective notebook computers, the terminal module Hardware Socket 72 transmits an infra-red pulse sequence corresponding to the token "10", which in turn corresponds to the ACTIVE_OPEN request, for example, by direct modulation of the infra-red emitter in the input/output interface 69, to the infra-red receiver in the input/output interface 66 of the terminal module 48. The Hardware Socket 65 demodulates the received signal and passes the token to the Sockets Interface 64 which regenerates the ACTIVE_OPEN request from its look-up table and passes it to the TCP/IP module 63 (s14).

Once the ACTIVE_OPEN command is received by TCP/IP, it issues a connection identifier and sends an OPEN_ID message back to the application module FTP 82 over the infra-red connection (s15 - s17) by the token coding-decoding process described above, before seeking to initiate the connection to the server (s18 - s22). The way in which the TCP seeks to set up a connection-oriented data transfer path between the terminal module 48 and the remote server 81 connected to the Internet (s18 - s22) is entirely conventional and is described above in relation to Figure 5.

As well as the tokenised form of the commands, parameters may be passed using similar encoding principles.

Data transfer from the terminal module 48 to the application module 45 may take place while data is being transferred from the server, as soon as blocks of error-free data are available. Alternatively, the terminal module 48 may be provided with a substantial memory capability, for example, the integral hard disk on a notebook computer, in which data from the server is stored for later transmission to the application module 45.

The Hardware Socket software may operate with any sort of device which allows data to be transmitted serially or in parallel. In a typical notebook computer, the Hardware Socket may control a serial or parallel port as well as the I-R port. In each case, the Hardware Socket is implemented by software code which converts tokens from the Sockets Interface into a form which can be transmitted over an appropriate hardware device, and vice-versa.

The terminal module 48 may of course be a proprietary unit rather an existing device, built to meet the requirements specified above without the additional features such as a keyboard provided as standard in existing devices such as notebook or palmtop computers. The Sockets Interface and Hardware Socket are implemented in software to interface with a commercially available TCP/IP stack.

The application modules 45, 46, 47 may be similarly implemented with the minimal functionality required, so as to reduce cost. Conventional processing circuitry is provided together with operating software to run required applications and to interface with the Hardsock and Dummy Sockets Interface modules in memory. The operating software may be a system such as ELKS, a freely available cut-down version of UNIX which is designed to run on low power processors. A conventional data interface is required to communicate with the terminal module 48. This is implemented, for example, using a USART chip such as the Intel 8251A available from Maplin Electronics, UK. This is microprocessor programmable to provide a serial data input/output conforming to the standard RS-232 protocol, which can be connected to the terminal module directly or via an infra-red transceiver.

As with software sockets, a number of logical paths may be opened between the Hardware Socket modules to different application modules, for example, where a number of application modules 45, 46, 47 are simultaneously within infra-red range of the terminal module 48. Transmission/reception to and from each device may be on a simple round-robin basis. In a software sockets environment, when a socket is opened, a unique number is returned to be stored by the application for future use of the socket, and for the operating system to maintain the context for that socket. This identifier is known generically as a socket handle. Such a handle can be used in relation to the present invention, with the unique handle being sent as a tokenised parameter.

In a particular embodiment of the invention, the terminal module 48 comprises a plug-in PCMCIA-style card which includes an infra-red transmitter/receiver, a processor and memory, with operating software such as the ELKS system together with the Sockets Interface and Hardware Socket being implemented in software in the body of the card, so that connection is by plugging the card into a corresponding PCMCIA slot in the application module 47, with appropriate driver software implemented in the Hardware Socket 72 of the application module.

Referring to Figure 9, the system may be used within an office to keep users appraised of information relevant to their needs. For example, the device in the form of a roaming unit with integral card 48 is carried with the user at all times in moving around a large office. Infra-red transmitters 90 are mounted in the ceiling 91 from which data is transmitted to the card 48 when the user is within range. That data is stored within the terminal module memory until the user requires to access it, when he or she may approach a standard application module terminal 92 and retrieve the information by plugging the card into a corresponding slot on the terminal 92.

When the user moves outside the office environment, the device may switch to a radio frequency mode in which it continues to receive data targeted at it. Such data may then be read by plugging the card into a slot provided on, for example, a palmtop computer, from which messages may also be sent back to the network.

Although the invention has been illustrated by reference to a TCP/IP protocol stack based on the Internet, the principles are applicable to other networks and internetworks, whether using the TCP/IP stack or other communications protocols, such as the OSI Internet Protocol.

Claims

1. A communications system for mobile data transfer, comprising:
a communications network (25);
an application module (45, 46, 47) capable of processing network data; and
a separate terminal module (48) configured to provide a network independent data transport service to the application module.

2. A system according to claim 1, wherein, as between the application module and the terminal module, only the terminal module includes a protocol stack (12, 13).

3. A system according to claim 1 or 2, wherein the data format for data transfer between the network and the terminal module is different to that between the terminal module and the application module.

4. A system according to any preceding claim, wherein data transfer between the network and the terminal module is independent of data transfer between the terminal module and the application module.

5. A system according to any preceding claim, wherein the application module is configured such that data transfer between the network and the application module can only take place via the terminal module.

6. A system according to any preceding claim, including:
first means (50, 51, 52) for connecting the terminal module to the network; and
second means for connecting the terminal module to the application module.

7. A system according to claim 6, wherein the first connecting means includes an infra-red link (51).

8. A system according to claim 6 or 7, wherein the second connecting means includes an infra-red link.

9. A system according to any one of claims 6 to 8, wherein the second connecting means includes a card (49) configured to be connected in a corresponding slot in the application module.

10. A system according to any one of claims 1 to 5, wherein the terminal module comprises a card (48) configured to be plugged in to a corresponding slot in the application module.

11. A system according to any one of the preceding claims, wherein the terminal module and the application module each include means configured to establish a logical connection between them (64, 65, 66, 69, 71, 72).

12. A system according to claim 11, wherein the connecting means comprises a hardware socket (65, 72).

13. A terminal module (48) for a mobile communications network (25), comprising:

a first connecting means (50, 51, 52) for connecting the terminal module to the network; and

a second connecting means for connecting the terminal module to a separate application module (45, 46, 47);

wherein the terminal module is configured to provide a network independent data transport service to each of a plurality of separate application modules.

14. A terminal module according to claim 13, wherein the first connecting means includes an infra-red transmitter and receiver.

15. A terminal module according to claim 13 or 14, wherein the second connecting means includes an infra-red transmitter and receiver.

16. A terminal module according to claim 13, comprising a PCMCIA-style card.

17. A method of transferring data to an application module (45, 46, 47) in a communications network, comprising:
receiving data destined for the application module at a terminal module (48) separate from the application module; and
processing the received data so as to provide a network independent data transport service for the application module.

18. A communications system substantially as hereinbefore described with reference to the accompanying drawings.

19. A terminal module substantially as hereinbefore described with reference to the accompanying drawings.

20. A method of transferring data to an application module in a communications network substantially as hereinbefore described with reference to the accompanying drawings.

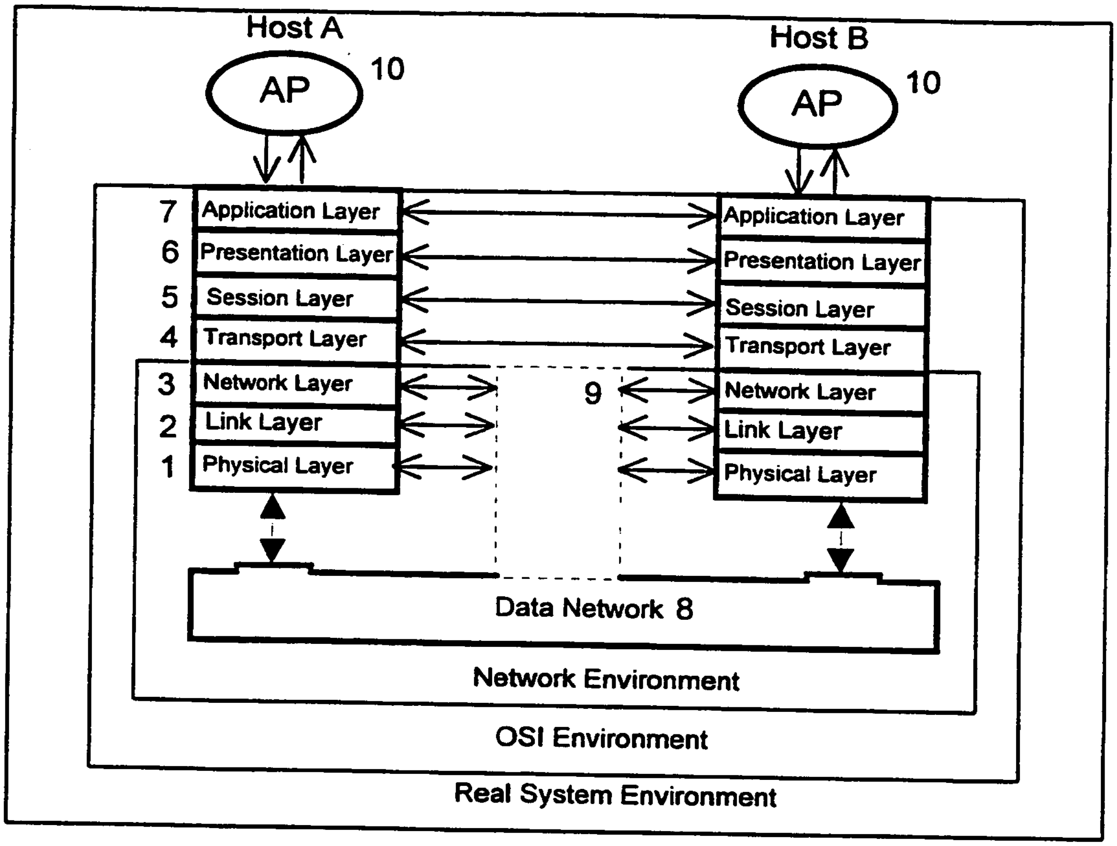


Figure 1

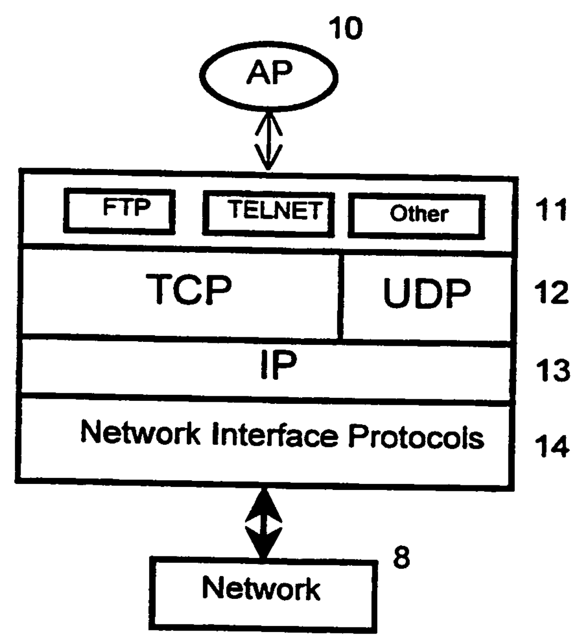


Figure 2

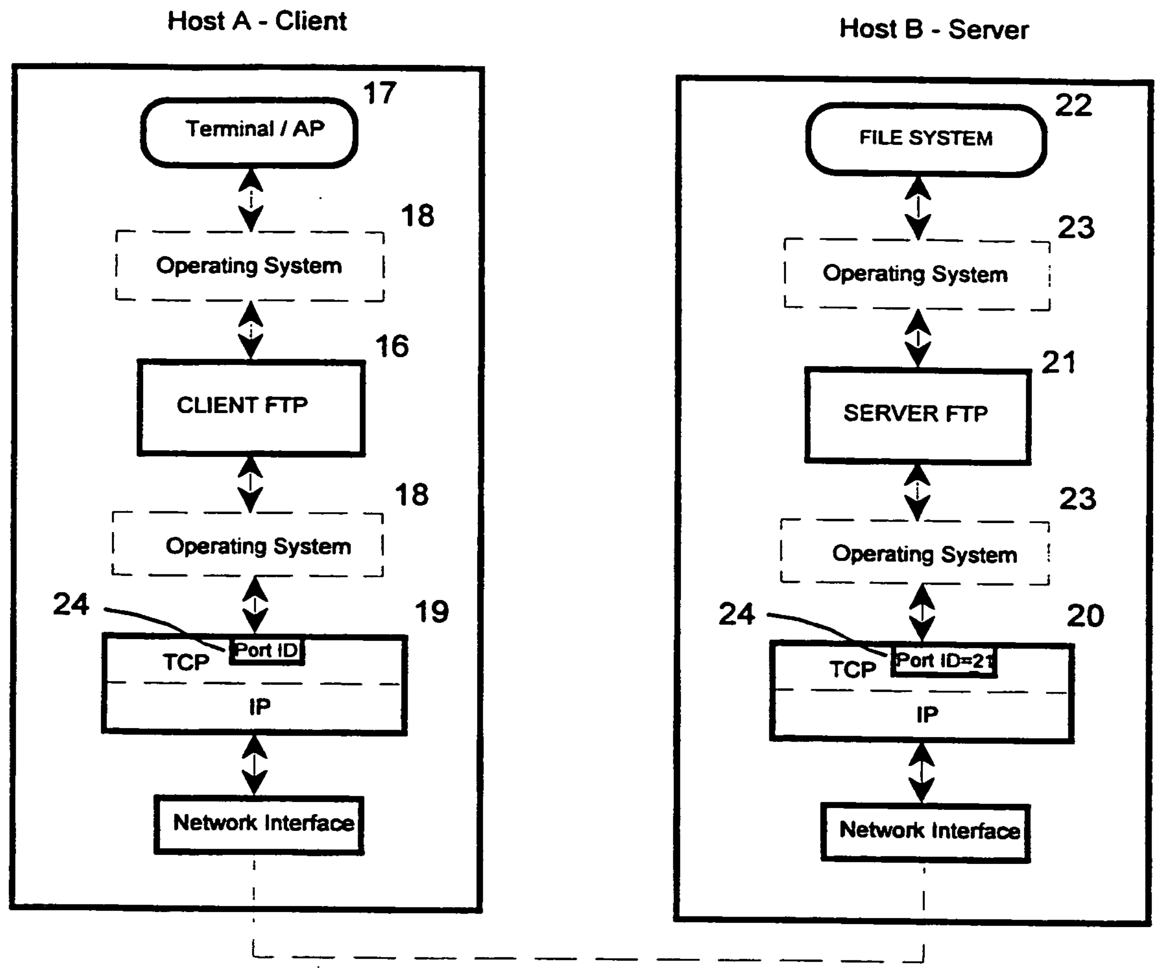

Host A - Client
Host B - Server
Terminal / AP
17
18
Operating System
16
CLIENT FTP
18
Operating System
24
19
TCP
Port ID
IP
Network Interface
FILE SYSTEM
22
23
Operating System
21
SERVER FTP
23
Operating System
24
20
TCP
Port ID=21
IP
Network Interface

Figure 3

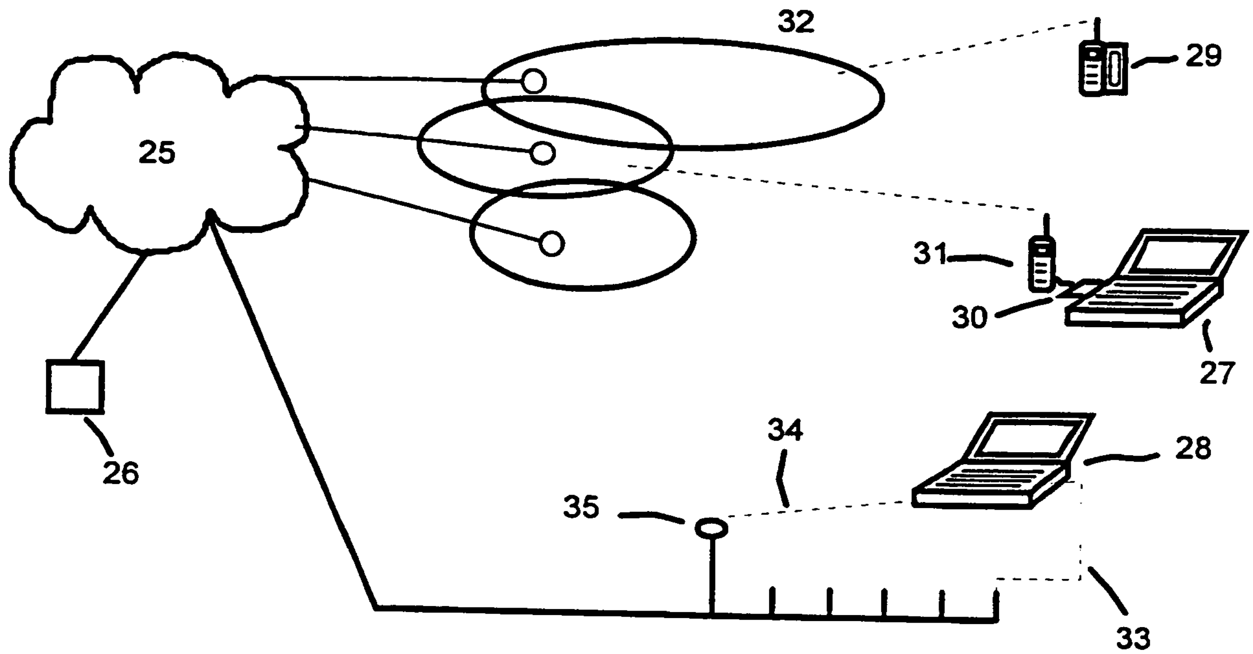


Figure 4

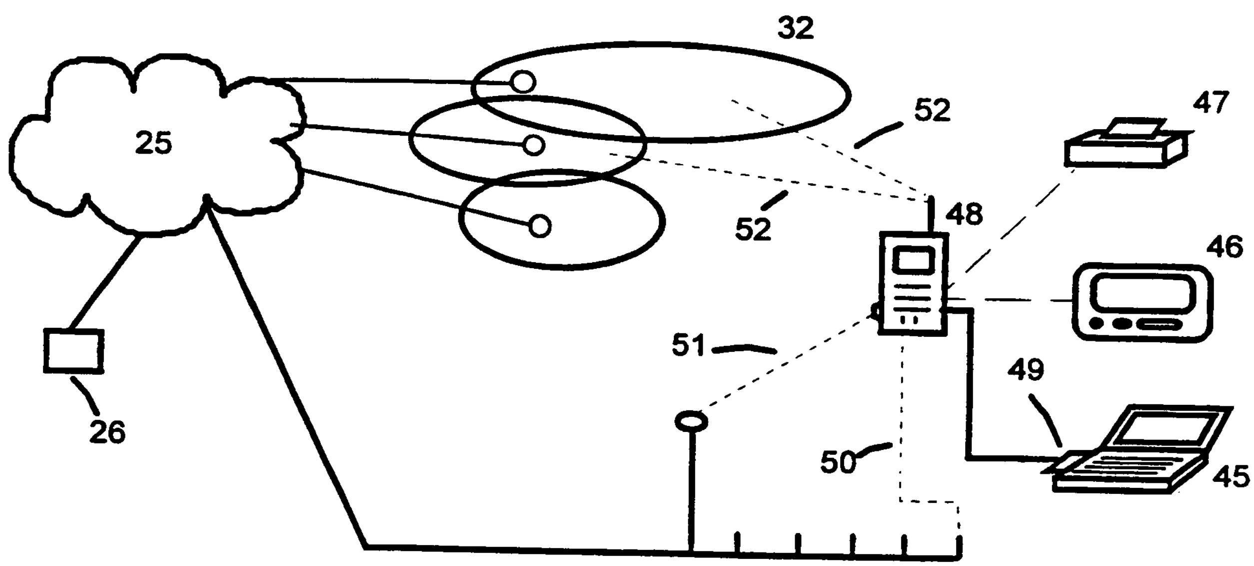


Figure 6

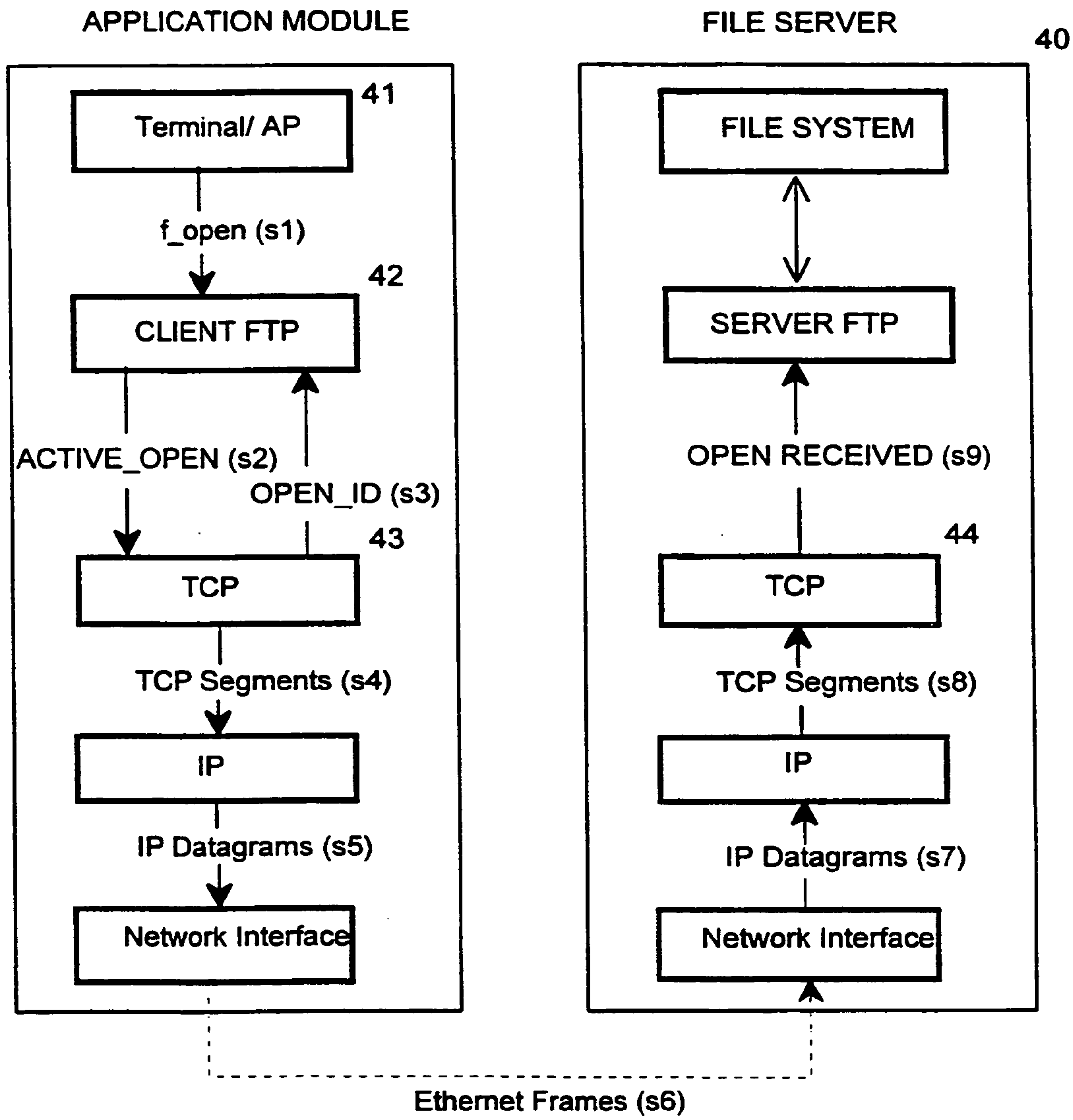

APPLICATION MODULE
FILE SERVER
40
Terminal/ AP
41
f_open (s1)
42
CLIENT FTP
ACTIVE_OPEN (s2)
OPEN_ID (s3)
43
TCP
TCP Segments (s4)
IP
IP Datagrams (s5)
Network Interface
FILE SYSTEM
SERVER FTP
OPEN RECEIVED (s9)
44
TCP
TCP Segments (s8)
IP
IP Datagrams (s7)
Network Interface
Ethernet Frames (s6)

Figure 5

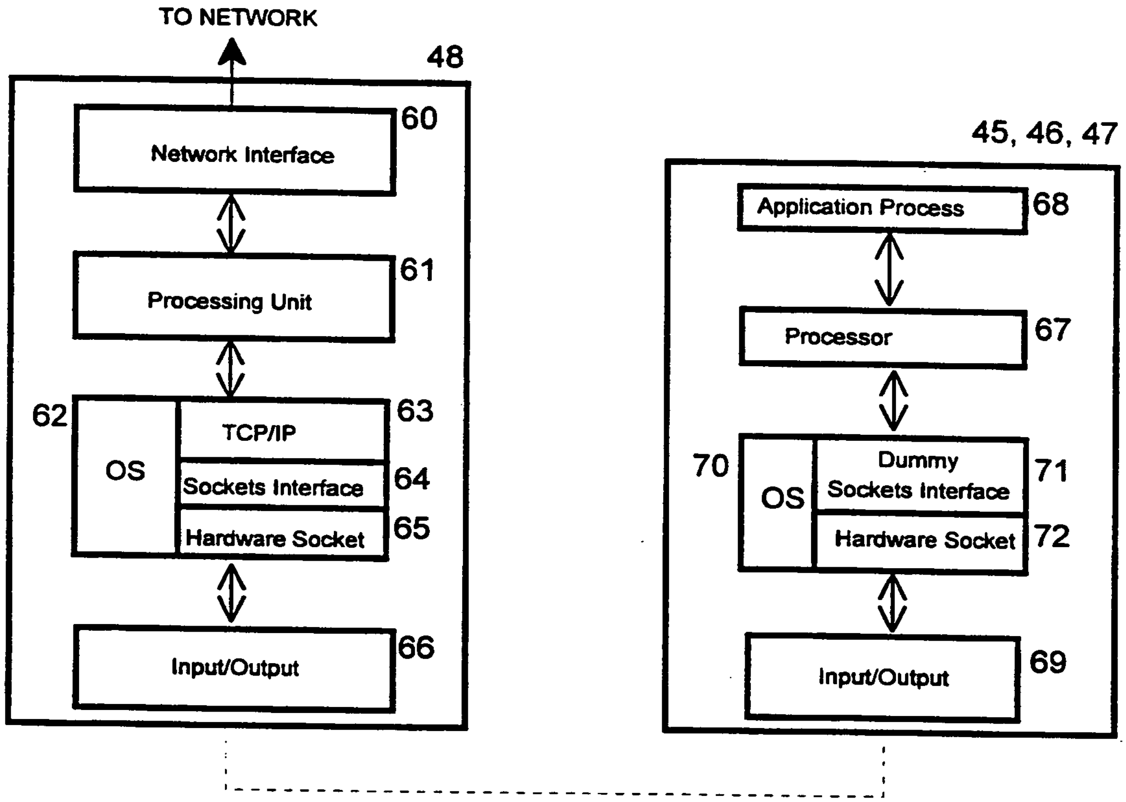

TO NETWORK
48
Network Interface
60
Processing Unit
61
62
OS
TCP/IP
63
Sockets Interface
64
Hardware Socket
65
Input/Output
66
45, 46, 47
Application Process
68
Processor
67
70
OS
Dummy Sockets Interface
71
Hardware Socket
72
Input/Output
69

Figure 7

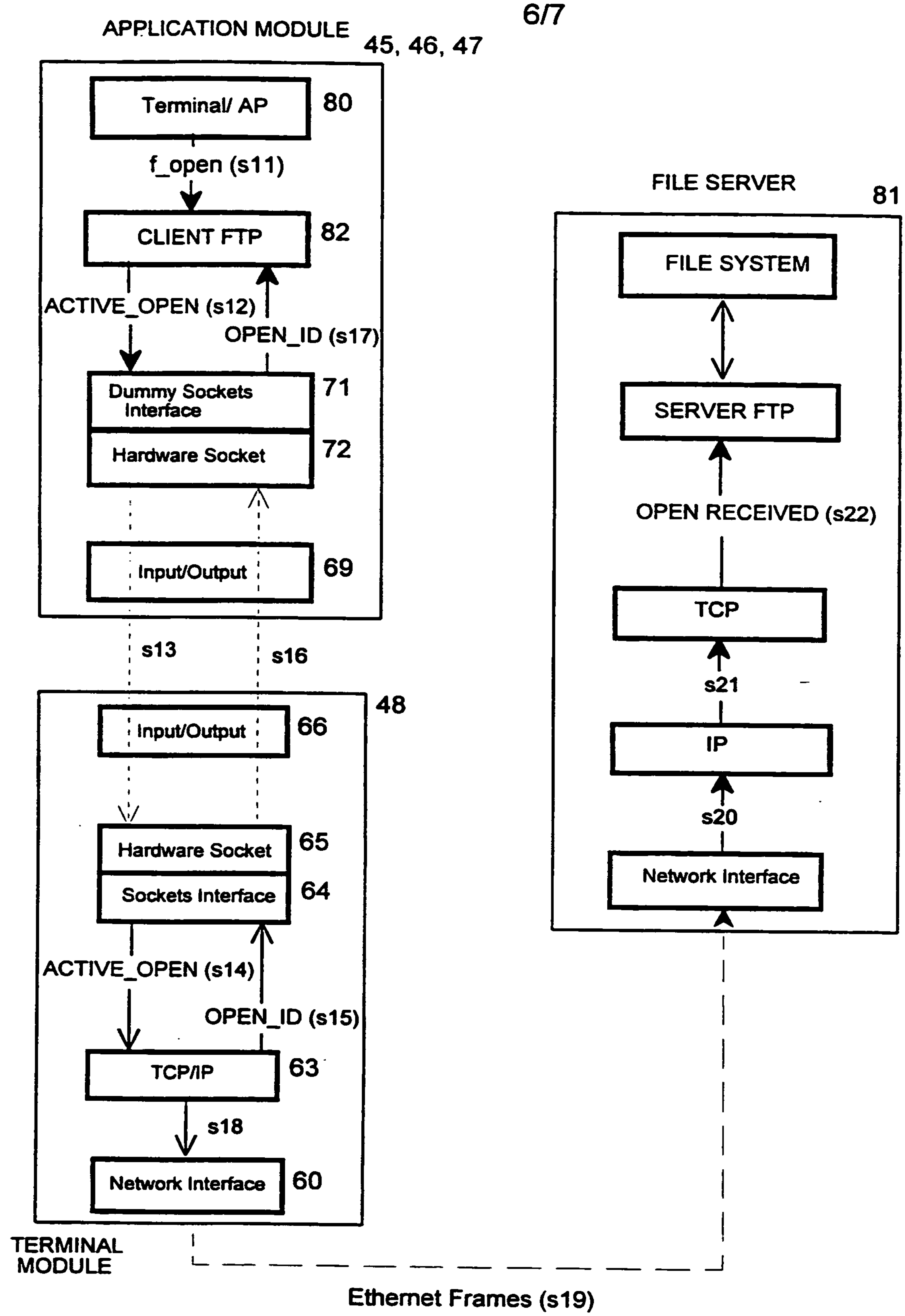

APPLICATION MODULE
45, 46, 47
Terminal/ AP
80
f_open (s11)
CLIENT FTP
82
ACTIVE_OPEN (s12)
OPEN_ID (s17)
Dummy Sockets Interface
71
Hardware Socket
72
Input/Output
69
s13
s16
48
Input/Output
66
Hardware Socket
65
Sockets Interface
64
ACTIVE_OPEN (s14)
OPEN_ID (s15)
TCP/IP
63
s18
Network Interface
60
TERMINAL MODULE
FILE SERVER
81
FILE SYSTEM
SERVER FTP
OPEN RECEIVED (s22)
TCP
s21
IP
s20
Network Interface
Ethernet Frames (s19)

Figure 8

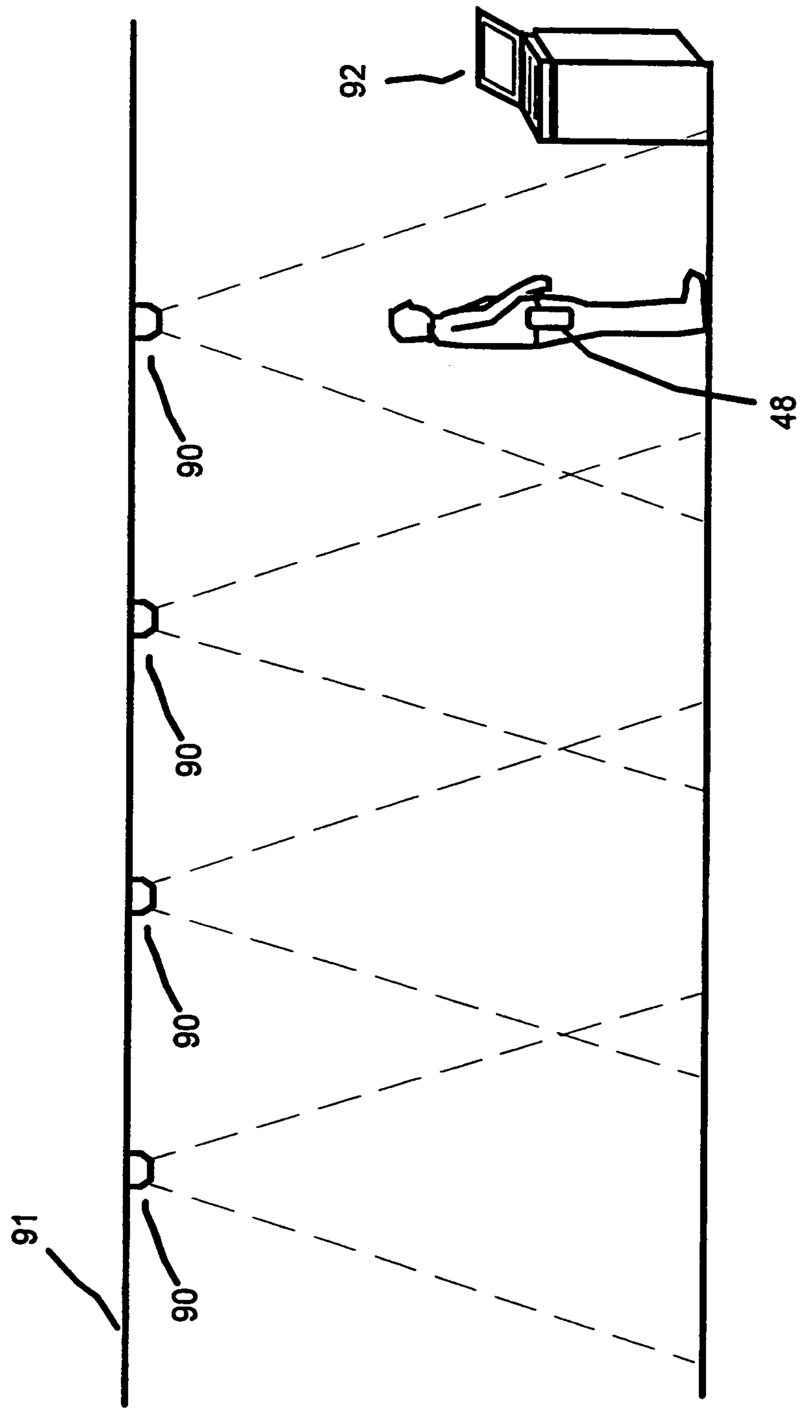


Figure 9

INTERNATIONAL SEARCH REPORT

International Application No
PCT/GB 99/00311

A. CLASSIFICATION OF SUBJECT MATTER
IPC 6 H04L29/08 G06F15/02 H04L29/06

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 6 G06F H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	EP 0 637 801 A (BULL SA) 8 February 1995 see abstract see page 2, line 1 - page 7, line 58 * annex 1-4 * see figures 1-3 ---	1,13, 17-20
A	EP 0 748 096 A (XEROX CORP) 11 December 1996 see abstract see page 1, line 1 - page 6, line 53 see figures 1,2 ---	1,13, 17-20
A	DE 196 18 535 A (SIEMENS AG) 24 July 1997 see the whole document --- -/--	1,13, 17-20

[X] Further documents are listed in the continuation of box C.

[X] Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
26 May 1999	07/06/1999

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Lievens, K

1

Form PCT/ISA/210 (second sheet) (July 1992)

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication,where appropriate, of the relevant passages	Relevant to claim No.
A	INFORMATION SCIENCES INTITUTE, UNIVERSITY OF SOUTHERN CALIFORNIA: "TRANSMISSION CONTROL PROTOCOL" RFC793,30 September 1981, XP002069478 cited in the application ftp://ds.internic.net/rfc/rfc791.txt see page 8, paragraph 2.3 see page 9, paragraph 2.4 see page 10-11, paragraph 2.7 see page 44-51, paragraph 3.8 see page 52-77, paragraph 3.9 -----	1,13, 17-20

1

INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No
PCT/GB 99/00311

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 0637801 A	08-02-1995	FR 2708116 A	27-01-1995
		WO 9503583 A	02-02-1995
EP 0748096 A	11-12-1996	JP 9016486 A	17-01-1997
DE 19618535 A	24-07-1997	NONE	

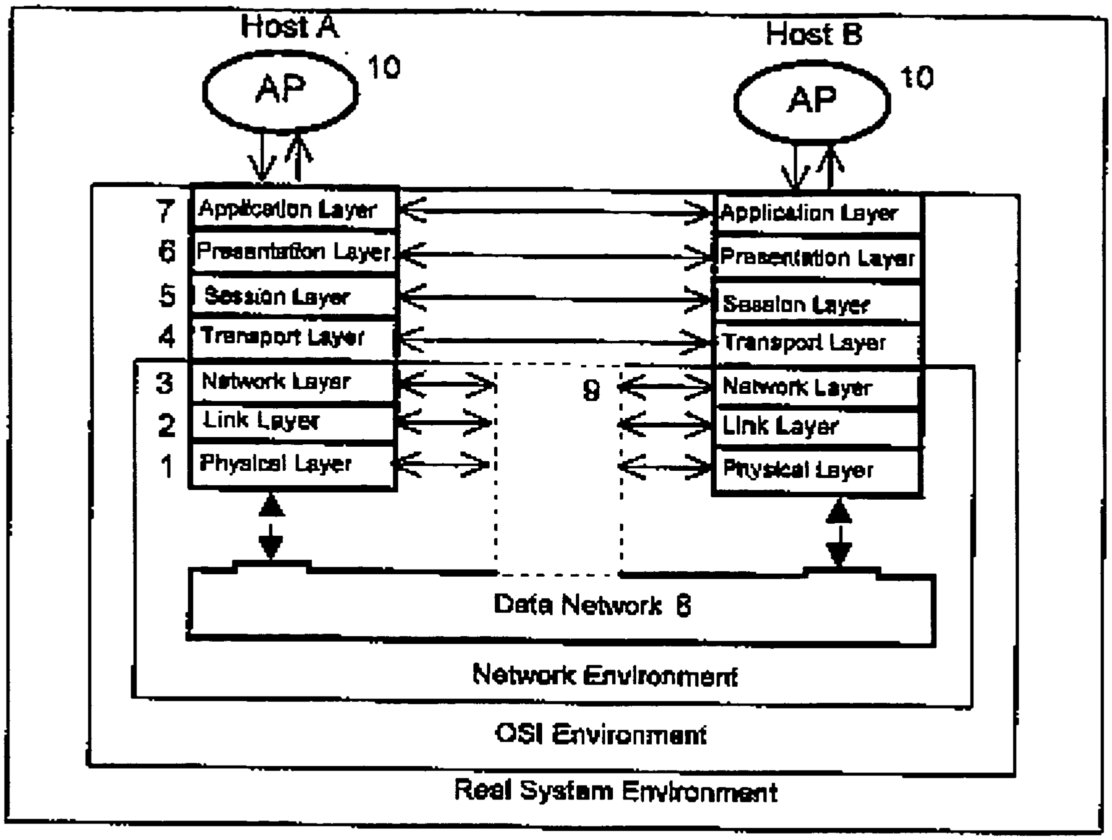


Figure 1

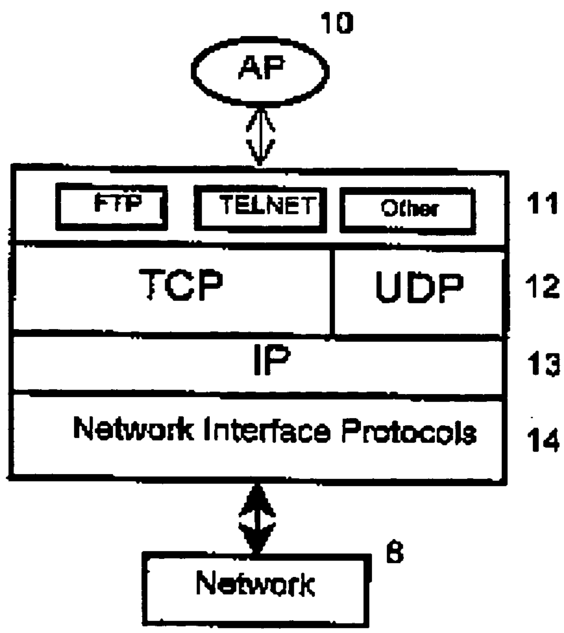
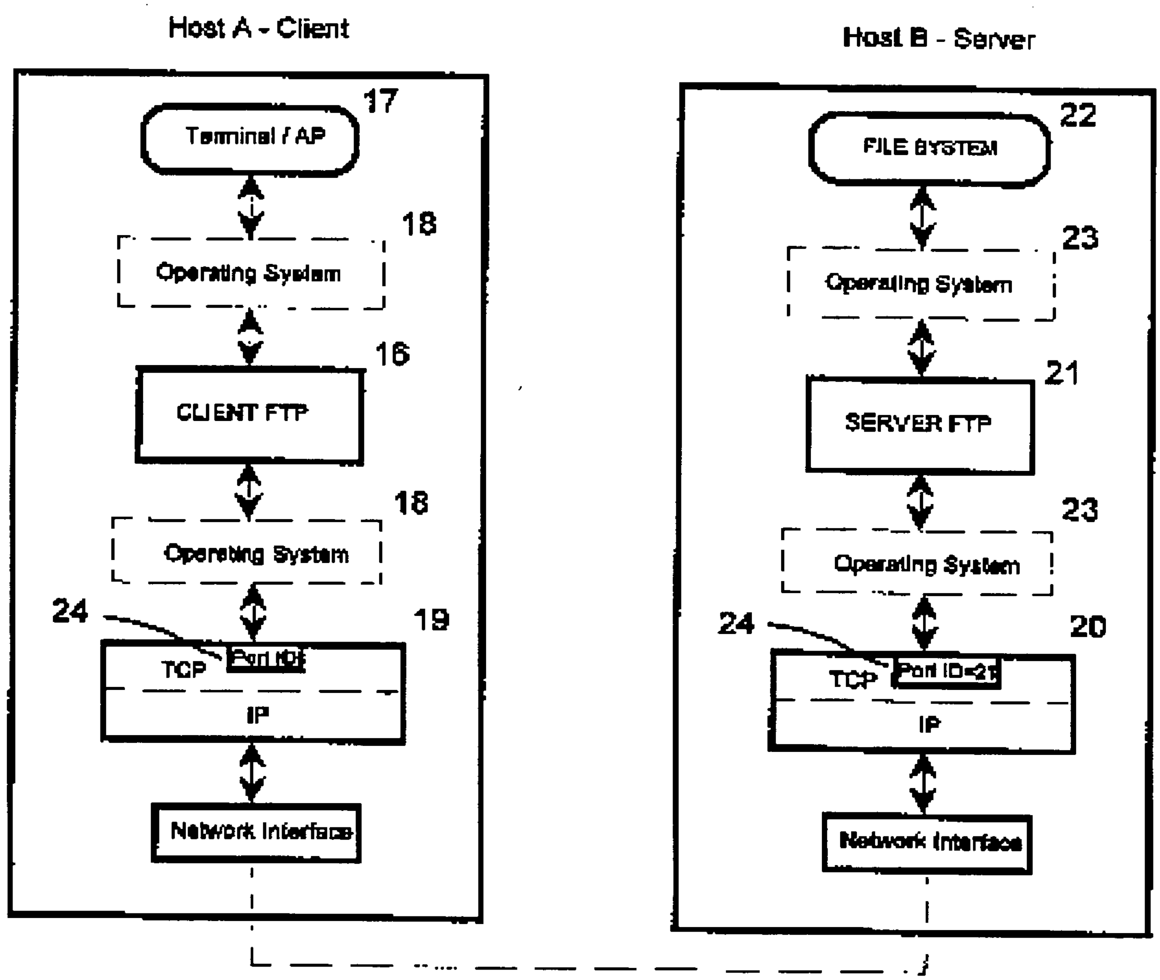


Figure 2



Host A - Client
Host B - Server
Terminal / AP
17
18
Operating System
16
CLIENT FTP
18
Operating System
24
19
TCP
Port ID
IP
Network Interface
FILE SYSTEM
22
23
Operating System
21
SERVER FTP
23
Operating System
24
20
TCP
Port ID=21
IP
Network Interface

Figure 3

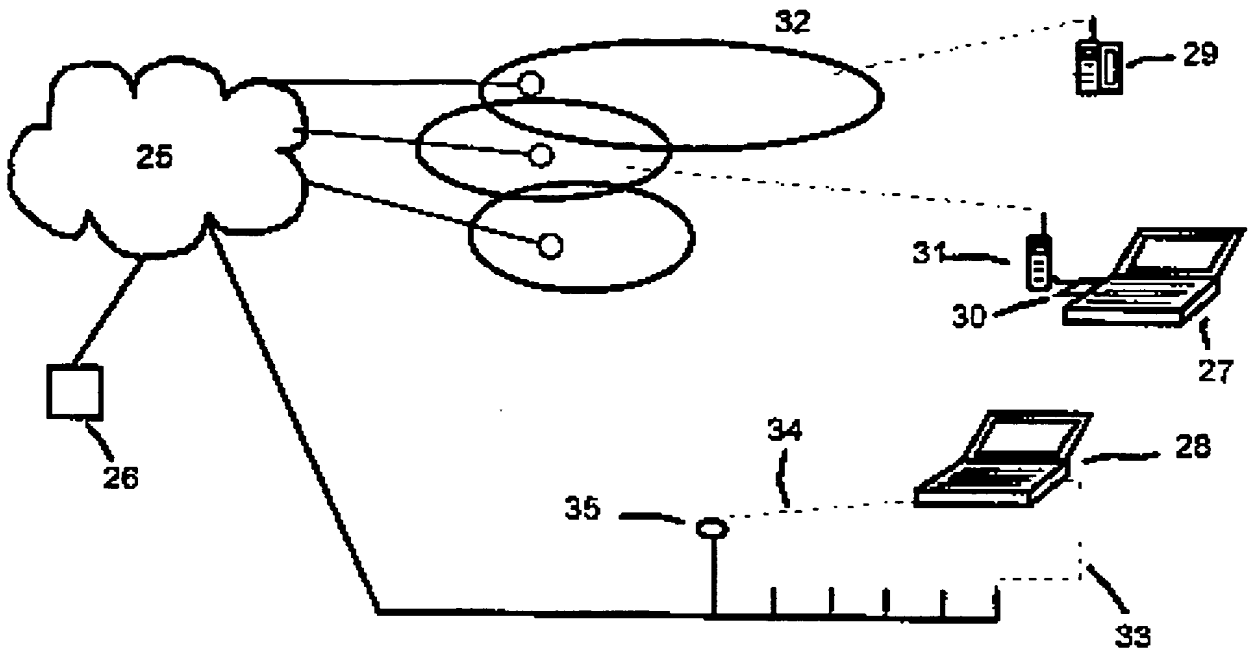


Figure 4

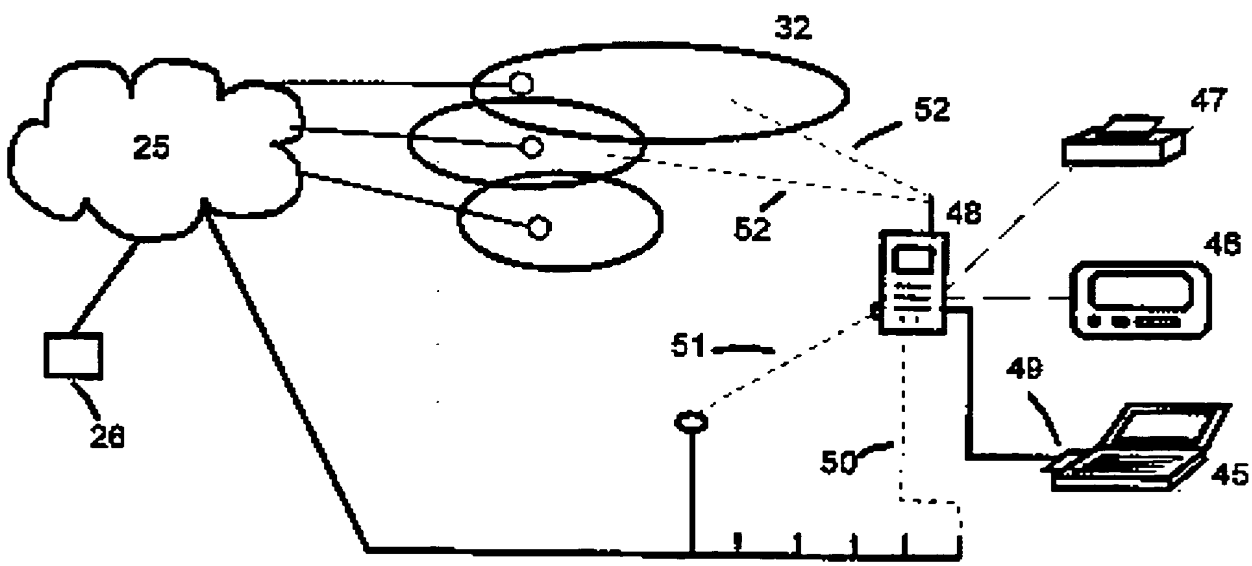


Figure 6

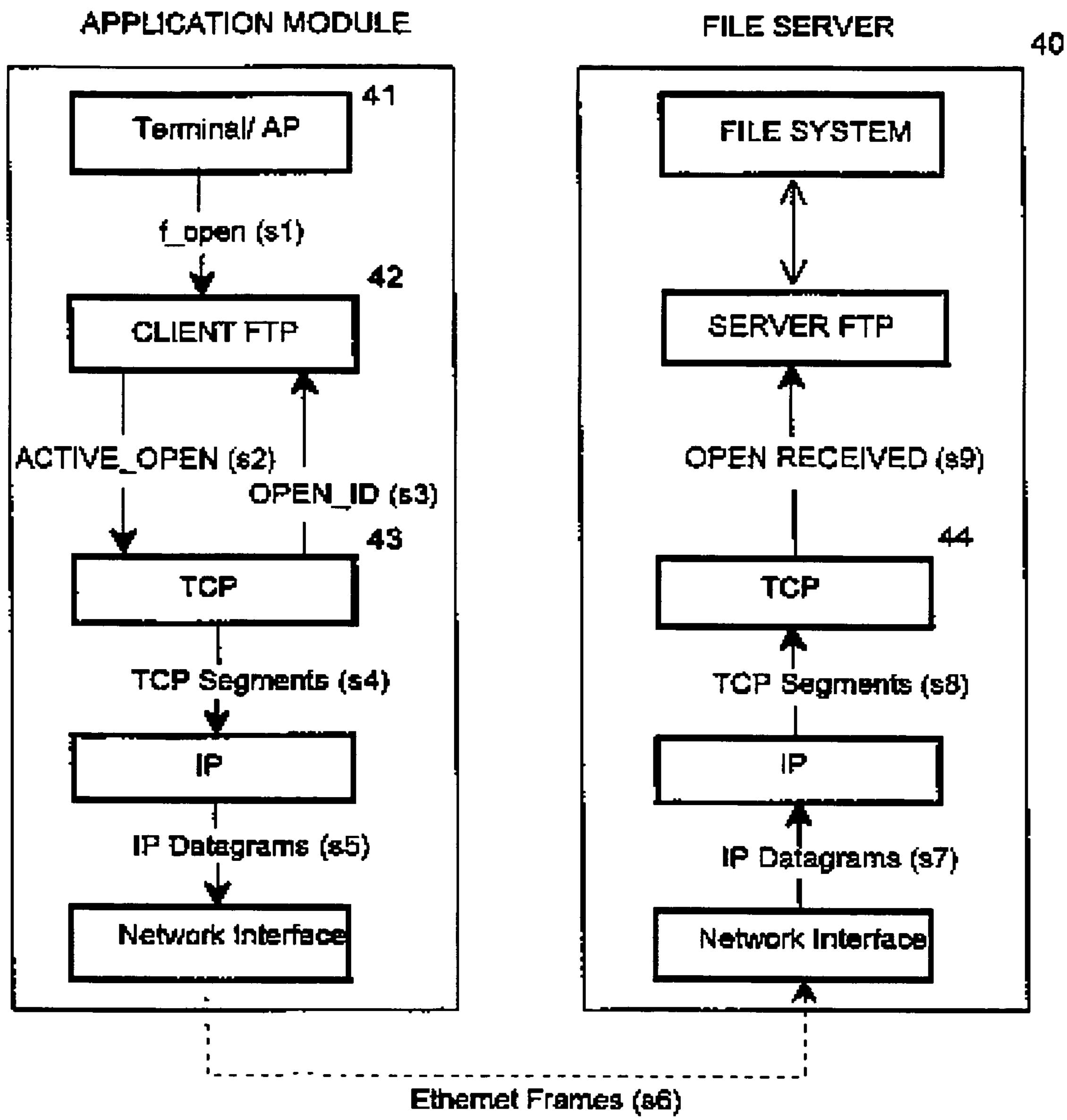

APPLICATION MODULE
FILE SERVER
40
41
Terminal/ AP
f_open (s1)
42
CLIENT FTP
ACTIVE_OPEN (s2)
OPEN_ID (s3)
43
TCP
TCP Segments (s4)
IP
IP Datagrams (s5)
Network Interface
FILE SYSTEM
SERVER FTP
OPEN RECEIVED (s9)
44
TCP
TCP Segments (s8)
IP
IP Datagrams (s7)
Network Interface
Ethernet Frames (s6)

Figure 5

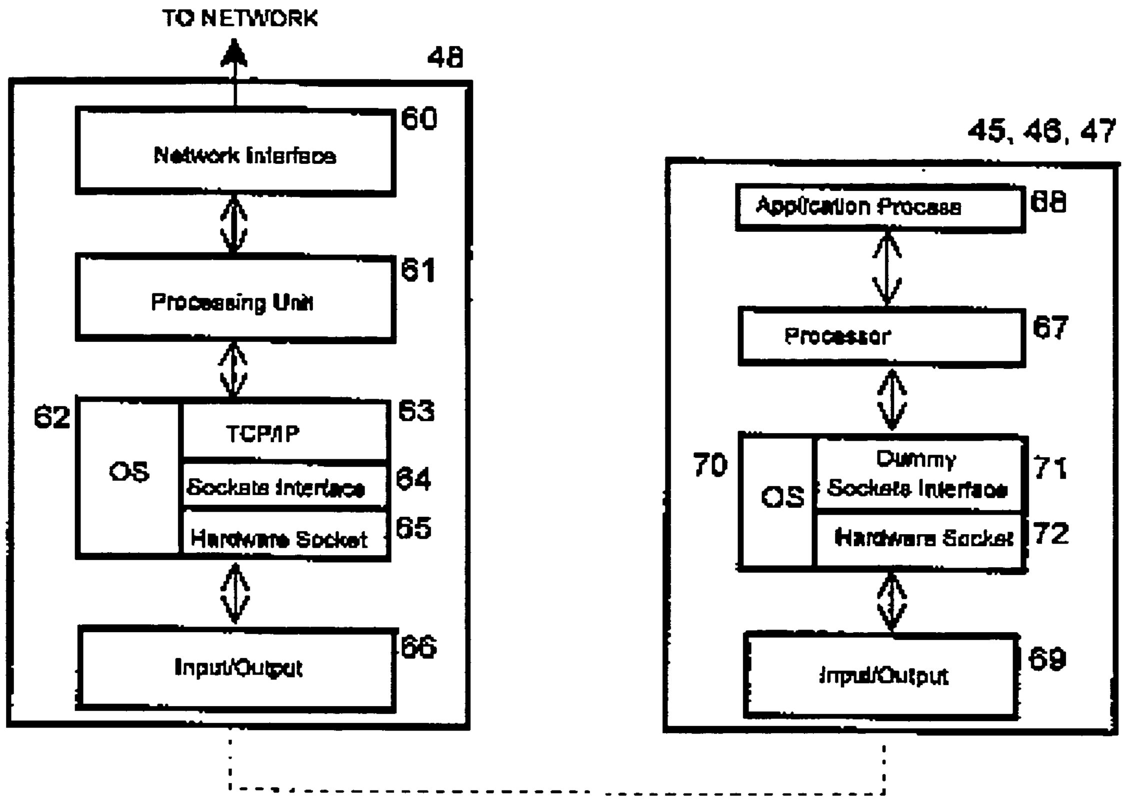

TO NETWORK
48
Network Interface
60
Processing Unit
61
62
OS
TCP/IP
63
Sockets Interface
64
Hardware Socket
65
Input/Output
66
45, 46, 47
Application Process
68
Processor
67
70
OS
Dummy Sockets Interface
71
Hardware Socket
72
Input/Output
69

Figure 7

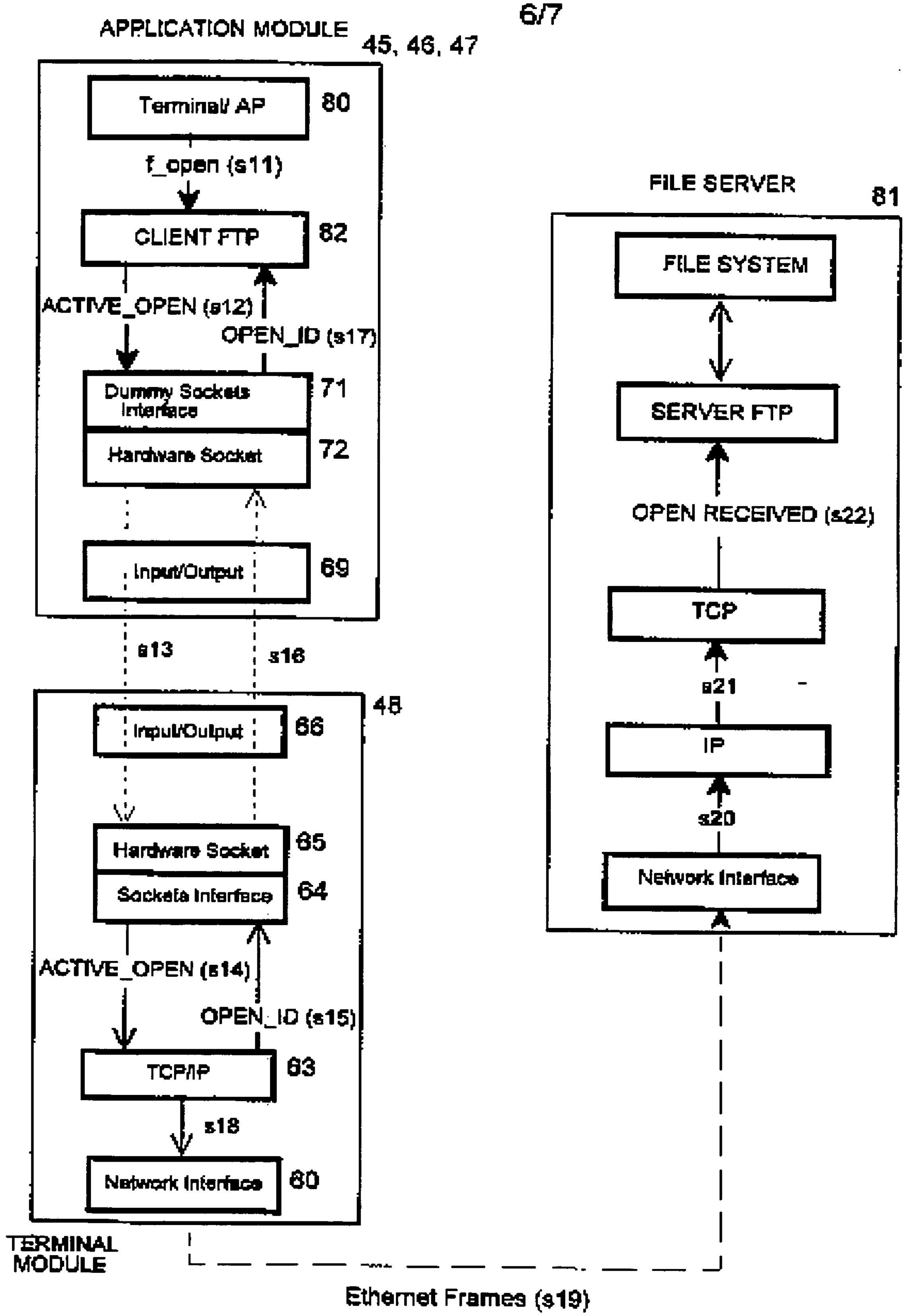

APPLICATION MODULE
45, 46, 47
Terminal/ AP
80
f_open (s11)
CLIENT FTP
82
ACTIVE_OPEN (s12)
OPEN_ID (s17)
Dummy Sockets Interface
71
Hardware Socket
72
Input/Output
69
s13
s16
48
Input/Output
66
Hardware Socket
65
Sockets Interface
64
ACTIVE_OPEN (s14)
OPEN_ID (s15)
TCP/IP
63
s18
Network Interface
60
TERMINAL MODULE
FILE SERVER
81
FILE SYSTEM
SERVER FTP
OPEN RECEIVED (s22)
TCP
s21
IP
s20
Network Interface
Ethernet Frames (s19)

Figure 8

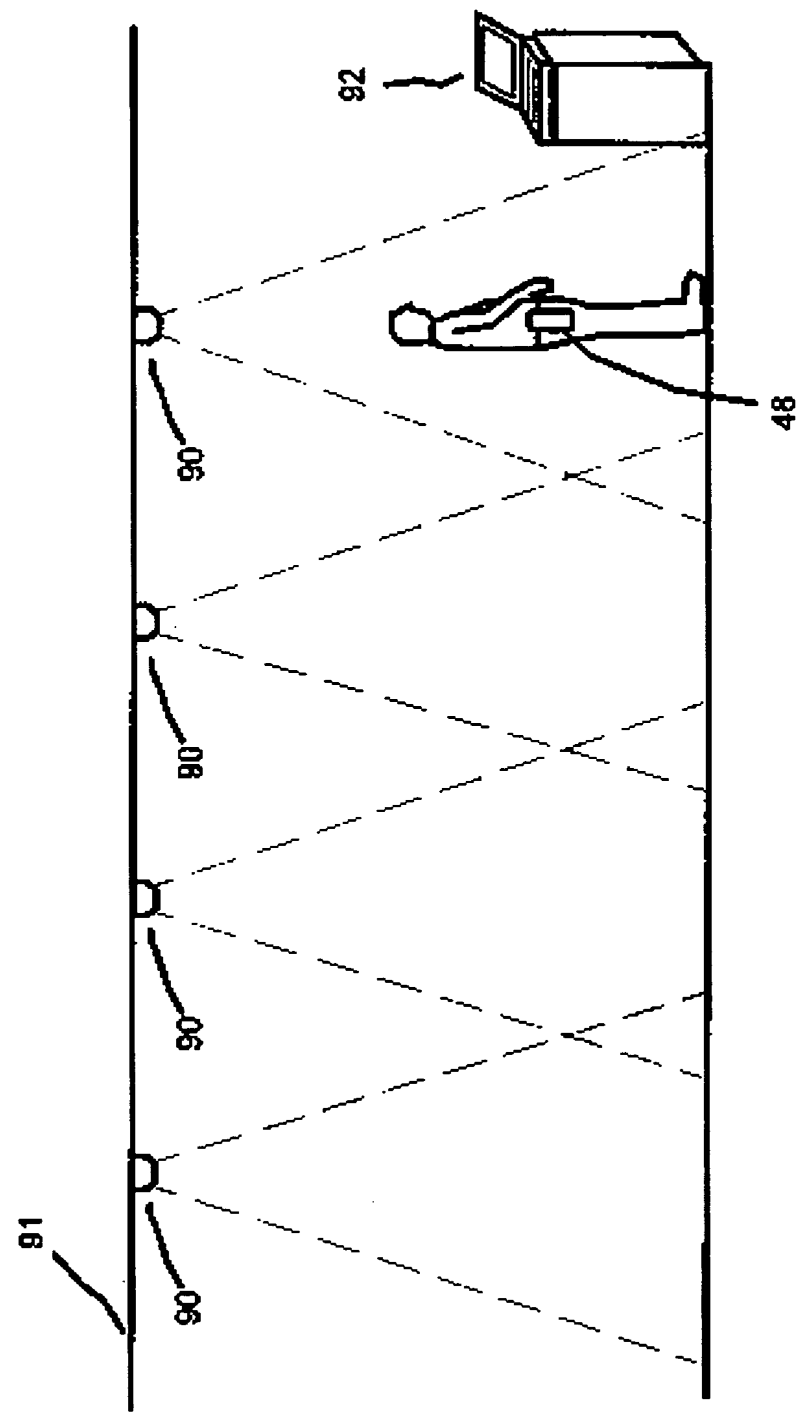


Figure 9